Accelerating Simulation of Population Continuous Time Markov Chains via Automatic Model Reduction

Accelerating Simulation of Population Continuous Time Markov Chains via Automatic Model Reduction

Cheng Feng[a,*], Jane Hillston[b]

[a] *Institute for Security Science and Technology, Imperial College London, London, UK*
[b] *LFCS, School of Informatics, University of Edinburgh, Informatics Forum, Edinburgh, UK*

Abstract

We present a novel model reduction method which can significantly boost the speed of stochastic simulation of a population continuous-time Markov chain (PCTMC) model. Specifically, given a set of predefined target populations of the modellers' interest, our method exploits the coupling coefficients between population variables and transitions with respect to those target populations which are calculated based on a directed coupling graph constructed for the PCTMC. Population variables and transitions which have high coupling coefficients on the target populations are exactly simulated. However, the remaining population variables and transitions which have low coupling coefficients can either be removed or approximately simulated in the reduced model. The reduced model generated by our approach has significantly lower cost for stochastic simulation, but still retains high accuracy on the statistical properties of the target populations. The applicability and effectiveness of our method is demonstrated on two illustrative models.

Keywords: Population continuous time Markov chain; Stochastic simulation; Model Reduction

*Corresponding author
Email addresses: c.feng@imperial.ac.uk (Cheng Feng), jane.hillston@ed.ac.uk (Jane Hillston)

1. Introduction

Population models which consist of a large number of interacting individuals each belonging to a particular population, have been widely used to study many dynamic systems in different areas such as biology [1], ecology [2], the spread of epidemics [3], chemical reaction networks [4], and more recently, smart transportation like public biking-sharing systems [5, 6, 7]. These can considered with either continuous or discrete time, leading to models specified in terms of the rates or probabilities of events leaving a state, respectively. When exponentially distributed rates are assumed for the events within such models, the models are described as a subclass of Continuous Time Markov Chains (CTMCs) known as Population CTMCs (PCTMCs). Despite their usefulness, PCTMC models usually have a very large or even infinite state space, which has stimulated a lot of work in the computer science community to craft efficient algorithms for their analysis.

Fluid approximation techniques such as mean-field [8, 9] and moment approximation methods [10, 11] which approximate the large or infinite set of Chemical Master Equations (CMEs) describing the probability distributions of the populations over time by a much smaller set of ordinary differential equations (ODEs), can provide a means of rapid analysis of some specific metrics such as the mean, variance, and possibly other higher order moments of the populations in the models. However, when more sophisticated statistical properties need to be checked, a more generic and informative transient analysis is needed. This is often achieved by employing statistical model checking based on the Stochastic Simulation Algorithm (SSA) [12]. Specifically, the SSA is an exact method to numerically solve the CMEs by simulating a large number of possible trajectories of the model. Although the SSA is able to simulate any PCTMCs, in practice, the inefficiency of SSA has become an obstacle for many realistic models [13]. As a result, numerous approaches have been proposed to improve the efficiency of SSA, ranging from many accelerated variants [14, 15] to approximate methods such as tau-leaping [16, 17]. Since these methods have

to simulate either all or a substantial part of the transition events within the models, they are still not scalable enough for large models, i.e. models with large populations, a large number of populations or populations in which individuals have a large set of behaviours. A much more efficient approach is to exploit the presence of different time scales for model reduction [18, 19, 20, 21, 22]. However, a downside for this approach is that the identification of separable time scales is usually a manual process which is expensive and error-prone.

In this paper, we propose a novel approach which can significantly speed up the stochastic simulation of PCTMCs without any prior knowledge of model dynamics. This is achieved by automatically generating a reduced version of the original PCTMC in which the key dynamics of the model are preserved. Concretely, our method is based on the assumption that the modeller is only interested in checking the statistical properties of a few target populations in the PCTMC (which is often the case for statistical model checking). More specifically, we first define a directed coupling graph for an arbitrary PCTMC which quantifies the coupling between population variables and transitions in the PCTMC. By utilizing the moment approximation technique, the graph can be constructed at a relatively low computational cost compared with the total stochastic simulation cost. Then, given a few user-defined target populations, we propose a graph-based decoupling algorithm which splits the PCTMC into a pivotal part and a trivial part based on their coupling with respect to those target populations. Transitions in the pivotal part are those which have significant impact on the evolution of target populations, thus these must be exactly simulated. However, transitions in the trivial part are those whose impact is negligible or minor, thus these can be either discarded from the simulation or approximately simulated without causing significant deviation in the evolution of the target populations. Specifically, for the transitions in the trivial part, we define border transitions as those which can directly influence any population variables in the pivotal part of the decoupled PCTMC. Then, the coupling of the trivial part to the pivotal part of the decoupled PCTMC can be truncated by approximating the firing rates of those border transitions by state-independent

constants capturing their average behaviour, resulting in the removal of all the other transitions and population variables in the trivial part as they no longer influence the evolution of the target populations. Since all border transitions are selected to be those which have rather limited impact on the target populations, the approximation of their rates will be unlikely to cause significant deviation in the target populations. Furthermore, we also propose a moment approximation-based validation method to check the accuracy of the reduced model before stochastic simulation runs are conducted. We demonstrate the effectiveness of our method by applying it to two illustrative PCTMC models. The result shows that the reduced models generated by our method can significantly bring down the cost of stochastic simulation runs but still produce accurate statistical metrics on the target populations compared with the original models.

Thus the contribution of this paper is a novel model reduction technique for PCTMCs analysed via simulation. We present algorithms for automatically carrying out the model reduction according to a user-specified decoupling threshold and for validation of the model with respect to an error threshold caused by the reduction. This represents a substantial modification of our preliminary work on this topic, which was presented in [23]. The current work makes a significant improvement in the efficiency with which the number of transition firings is estimated. In [23] we eliminated the non-influential parts of the model entirely; now we retain *border transitions* so that the influence of the trivial part of the model is not completely lost, improving the faithfulness of the reduced model. Moreover a novel validation technique for estimating the accuracy of the reduced model is presented.

The outline of the paper is summarised as follows. The next section will give the background on the PCTMCs, their moment approximation and stochastic simulation algorithms for analysing such models. Then, we present the decoupling method which splits a PCTMC into the pivotal part and trivial part in Section 3. This is followed by the description of the reduction method for the generation of the reduced version of the original PCTMC based on its decou-

pled representation in Section 4. Section 5 describes how to validate the reduced model before it is used for simulation. Section 6 gives the two case studies which illustrate the applicability and power of our method for accelerating stochastic simulation of PCTMCs. Finally, the last section presents our conclusion.

2. Background

2.1. PCTMC

A population continuous time Markov chain (PCTMC) is a stochastic process whose states are captured by a numerical population vector, and transitions between states are defined by changes in some of the populations, with exponentially distributed rates expressed as functions of populations. The analysis of interest for such models is often the evolution of different populations over time. Formally, a PCTMC can be represented as a tuple $\mathcal{P} = (\mathbf{x}, \mathcal{T}, \mathbf{x}_0)$, where:

- $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{Z}^n_{\geq 0}$ is an integer vector with the ith $(1 \leq i \leq n)$ component representing the current population of an agent type S_i. Each x_i takes values in a finite domain $\mathcal{D}_i \subseteq \mathbb{Z}_{\geq 0}$. Hence, $\mathcal{D} = \prod_{i=1}^{n} \mathcal{D}_i$ is the state space of the model.
- $\mathcal{T} = \{\tau_1, ..., \tau_m\}$ is the set of transitions, of the form $\tau = (r_\tau(\mathbf{x}), \mathbf{d}_\tau)$, where:
  1. $r_\tau(\mathbf{x}) : \mathbb{Z}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ is the rate function, associating with each transition the rate of an exponential distribution, depending on the global state of the model.
  2. $\mathbf{d}_\tau = (d^1_\tau, \ldots, d^n_\tau) \in \mathbb{Z}^n$ is the update vector which gives the net change for each population variables in $\mathbf{x}$ caused by transition τ.
- $\mathbf{x}_0 \in \mathbb{Z}^n_{\geq 0}$ is the initial state of the model.

For readability, transitions in PCTMCs can be expressed in the chemical reaction style with S_i being a specific molecular species/population, $r_\tau(\mathbf{x})$ being

the reaction propensity function and $\mathbf{d}_\tau$ capturing the consumed and produced population of species by the reaction:

$$\underline{N_1}S_1 + \ldots + \underline{N_n}S_n \quad \longrightarrow_\tau \quad \overline{N_1}S_1 + \ldots + \overline{N_n}S_n \quad \text{at rate } r_\tau(\mathbf{x})$$

where the net change on the population of agent type/species S_i due to transition τ is given by $d^i_\tau = \overline{N_i} - \underline{N_i}$ $(1 \leq i \leq n)$.

The probability distribution of the populations over time is given by the Chemical Master Equation (CME):

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{P}(\mathbf{x}, t \mid \mathbf{x}_0) = \sum_{\tau \in \mathcal{T}} [r_\tau(\mathbf{x} - \mathbf{d}_\tau)\mathrm{P}(\mathbf{x} - \mathbf{d}_\tau, t \mid \mathbf{x}_0) - r_\tau(\mathbf{x})\mathrm{P}(\mathbf{x}, t \mid \mathbf{x}_0)]$$

where $\mathrm{P}(\mathbf{x}, t \mid \mathbf{x}_0, t_0)$ is the probability that $\mathbf{x}(t) = \mathbf{x}$ given initial state $\mathbf{x}(0) = \mathbf{x}_0$. Solving the above CME requires us to compute the solution of a differential equation for each possible state of the PCTMC. However, since the state space of PCTMCs are generally extremely large, it is infeasible to solve the CME for most cases.

2.2. Moment Approximation

Fortunately, if we are only interested in some particular moments (mean, variance, covariance, skewness, kurtosis, etc.) of the population dynamics in a PCTMC, we can solve a much smaller set of ODEs just for those moments. Specifically, let $M : \mathbb{R}^n_{\geq 0} \to \mathbb{R}$ be a moment function, then the moment described by M evolves according to the following differential equation [24]:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[M(\mathbf{x}(t))] = \sum_{\tau \in \mathcal{T}} \mathbb{E}[(M(\mathbf{x}(t) + \mathbf{d}_\tau) - M(\mathbf{x}(t)))r_\tau(\mathbf{x}(t))] \tag{1}$$

with $\mathbb{E}[M(\mathbf{x}(0))] = M(\mathbf{x}_0)$. For example, if we set $M(\mathbf{x}(t)) = x_i(t)$, $M(\mathbf{x}(t)) = x_i^2(t)$, $M(\mathbf{x}(t)) = x_i(t)x_j(t)$, we get the following ODEs to describe the first moment, the second moment and the second-order joint moment respectively,

of population variables in an arbitrary PCTMC:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[x_i] &= \sum_{\tau\in\mathcal{T}} \mathbb{E}[(x_i + d_\tau^i - x_i) r_\tau(\mathbf{x})] = \sum_{\tau\in\mathcal{T}} d_\tau^i \mathbb{E}[r_\tau(\mathbf{x})] && (2)\\
\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[{x_i}^2] &= \sum_{\tau\in\mathcal{T}} \mathbb{E}[((x_i + d_\tau^i)^2 - {x_i}^2) r_\tau(\mathbf{x})] \\
&= 2\sum_{\tau\in\mathcal{T}} d_\tau^i \mathbb{E}[x_i \times r_\tau(\mathbf{x})] + \sum_{\tau\in\mathcal{T}} {d_\tau^i}^2 \mathbb{E}[r_\tau(\mathbf{x})] && (3)\\
\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[x_i x_j] &= \sum_{\tau\in\mathcal{T}} \mathbb{E}[((x_i + d_\tau^i)(x_j + d_\tau^j) - x_i x_j) r_\tau(\mathbf{x})] \\
&= \sum_{\tau\in\mathcal{T}} d_\tau^i \mathbb{E}[x_j \times r_\tau(\mathbf{x})] + \sum_{\tau\in\mathcal{T}} d_\tau^j \mathbb{E}[x_i \times r_\tau(\mathbf{x})] + \sum_{\tau\in\mathcal{T}} d_\tau^i \times d_\tau^j \mathbb{E}[r_\tau(\mathbf{x})] && (4)
\end{aligned}$$

where we use x_i as short for $x_i(t)$, and $r_\tau(\mathbf{x})$ as short for $r_\tau(\mathbf{x}(t))$ for convenience. The above system of ODEs does not necessarily have a solution since the dynamics of lower-order moments can depend on higher-order moments if the rate of any transition is a nonlinear function of population variables. For example, if we let $r_\tau(\mathbf{x}) = x_i x_j$, then an infinite number of ODEs are required to describe moment dynamics. In order to deal with this problem, various moment-closure methods have been proposed in the literature to truncate the system of ODEs at a certain order of moment.

The most common method to close the moment ODEs is to make a particular distribution assumption of the population variables. For example, the normal moment closure method assumes that the population variables at each point in time are approximately multivariate normal and therefore all third and higher-order moments can be expressed in terms of means and covariances. This relationship is captured by Isserlis' theorem [25]: For $\mathbf{x}$ multivariate normal with mean μ and covariance matrix σ_{ij}, we have

$$\begin{aligned}
\mathbb{E}[(\mathbf{x}-\mu)^{(\mathbf{m})}] &= \mathbb{E}[(x_1-\mu_1)^{(m_1)} \cdots (x_n-\mu_n)^{(m_n)}] = 0 \quad \text{if } o(\mathbf{m}) \text{ is odd} \\
\mathbb{E}[(\mathbf{x}-\mu)^{(\mathbf{m})}] &= \sum\prod \mathbb{E}[(x_i-\mu_i)(x_j-\mu_j)] \qquad\qquad \text{if } o(\mathbf{m}) \text{ is even}
\end{aligned}$$

where the notation $\sum\prod$ means summing over all distinct ways of partitioning $1,\ldots,n$ into pairs of i,j, $o(\mathbf{m}) = m_1 + \ldots + m_n$. For example, we can approximate

$$\mathbb{E}[x_1 x_2^2] \approx 2\mathbb{E}[x_2]\mathbb{E}[x_1 x_2] + \mathbb{E}[x_1]\mathbb{E}[x_2^2] - 2\mathbb{E}[x_1]\mathbb{E}[x_2]^2$$

if multivariate normal distribution for population variables is assumed, which yields $\mathbb{E}[(x_1 - \mu_1)(x_2 - \mu_2)^2] = 0$.

Apart from the normal closure method, there exist many other closure methods such as the moment expansion and central moment truncation method [26], log-normal [27, 28], beta-binomial [29] and Possion [30] closures. In general, computing moments of population variables using moment-closure approximation is much more efficient than stochastic simulation. However when the modelled system exhibits complex behaviour such as oscillations, the numerical result tends to be worse. It has been shown that increasing the number of moments can improve the accuracy in these cases [26]. However, when higher moments are required, the system of ODEs can be too large and sometimes the resulting ODEs may also become very stiff which makes them hard to solve numerically [24].

2.3. Stochastic Simulation Algorithms

The most generic and informative technique to analyse a PCTMC is to use stochastic simulation to numerically compute individual realisations of the underlying stochastic process. The idea is simply Monte Carlo: if we sample enough realisations of the stochastic process, then the estimates of statistical properties of the stochastic process will eventually converge to their true values. Specifically, given a PCTMC $\mathcal{P} = (\mathbf{x}, \mathcal{T}, \mathbf{x}_0)$ and the end time of simulation t_e, a trace of the PCTMC $\mathbf{x}(t)$ for $t \leq t_e$ can be calculated by the Gillespie's SSA [12] shown in Algorithm 1.

We can repeatedly apply the above algorithm to compute a large number of traces in order to estimate any statistical properties of a PCTMC such as the distribution or moments of the population vector $\mathbf{x}$ at any time point $t \leq t_e$.

In principle, Gillespie's SSA is able to analyse all PCTMCs. However in practice, due to the fact that the cost of SSA increases with the population size as well as the number of transitions, the inefficiency of SSA can clearly become an obstacle for many realistic models. As a consequence, numerous approaches have been proposed to improve the efficiency of SSA, including the optimized

Algorithm 1 Gillespie's SSA

Require: $\mathcal{P} = (\mathbf{x}, \mathcal{T}, \mathbf{x}_0)$, t_e

Set $t = 0$, $\mathbf{x} = \mathbf{x}_0$,
while $t \leq t_e$ **do**
  Generate two random numbers α, β uniformly distributed in $(0, 1)$
  Compute $r = \sum_{\tau \in \mathcal{T}} r_\tau(\mathbf{x})$
  Compute the time when the next transition fires as $t + h$, where $h = \frac{1}{r} \ln[1/\alpha]$ {sampling from an exponential distribution with rate r}
  if $t + h > t_e$ **then**
    break
  end if
  Compute which transition fires at time $t + h$ by finding τ_j such that

$$\beta \geq \frac{1}{r} \sum_{i=1}^{j-1} r_{\tau_i}(\mathbf{x}) \quad \text{and} \quad \beta < \frac{1}{r} \sum_{i=1}^{j} r_{\tau_i}(\mathbf{x})$$

  Set $t = t + h$, $\mathbf{x} = \mathbf{x} + \mathbf{d}_{\tau_j}$
end while

direct method [14], the next reaction method [15], and the composition rejection algorithm [31]. However, all these approaches are exact methods which means they have to simulate every transition event, thus their acceleration is rather limited. The tau-leaping methods [16, 17] speed up SSA by firing multiple transitions during a selected time interval instead of firing one transition at each step in the SSA given that the transition rates remain relatively constant during the selected time interval. However, tau-leaping methods are still not efficient enough for large models since they still try to capture a substantial proportion of the transition events in the PCTMC. Other approximate approaches mostly focus on exploiting the presence of different time scales in the model [18, 19, 20, 21, 22]. The common idea behind these approaches is to construct abstracted models, by decomposing a model into a fast and a slow subsystem (in some cases, even more time scales can be considered, but the general decomposition idea is the same). The fast subsystem is assumed to reach an equilibrium state at a time scale which is much faster than the time scale of the slow subsystem. Hence, the fast subsystem does not need to be simulated once it reaches its equilibrium state, and the system dynamics are dominated by the slow subsystem which can be simulated solely based on the equilibrium state of the fast subsystem. However, a common downside for these approaches is that the identification of fast and slow subsystems is usually a manual process, and requires expert knowledge of the dynamic behaviour of the model. This process is expensive and error-prone which significantly hinders the usage of these approaches. Although some pioneering work has been done to automate the separation process by obtaining the knowledge of the time scales through some experimental simulation runs of the entire model [32, 33], the strong precondition of the existence of clearly separable multiple time scales is still an obstacle for using the approaches on general PCTMC models.

Instead of utilizing the possible existence of multiple time scales within the models, we seek a different approach to speed up stochastic simulation of PCTMCs through an automatic model reduction method exploiting the cou-

pling coefficients of the transitions and population variables with respect to a set of predefined target populations which the modellers are interested in. Similar to our work is the directed relation graph (DRG)-based methods for skeletal mechanism reduction for the simulation of hydrocarbon oxidation, where a graph-based model reduction approach is also used for removing unimportant species and reactions whose contribution to species of interest is negligible [34]. The approach has since been improved by researchers in the combustion research domain such as DRG with error propagation [35], DRG with sensitivity analysis [36], etc. Our work is inspired by the DRG-based methods, however, there are some key differences. First, the DRG-based methods are used to reduce deterministic models whereas our work is applied to stochastic simulation. Second, since our goal is to speed up stochastic simulation, our primary focus is on transition reduction instead of the species (population) reduction that is the focus of the DRG-based methods. Moreover, instead of simply discarding less important dynamics in the DRG-based methods which can cause significant error, our approach is more robust since we still keep those less important dynamics but simulate them in a much less costly way. Lastly, although the DRG-based methods work well in the combustion simulation domain, they are still heuristics since no accuracy can be guaranteed for reductions. Our work has a validation step where the error on the target population dynamics can be estimated before it is used for future simulation, which makes our work more convincing and easier to apply.

3. The Decoupling Method

The model decoupling method is used to split a PCTMC into the pivotal part and the trivial part. Specifically, the pivotal part consists of transitions and population variables which have significant impact on the evolution of the target populations, whereas the trivial part consists of the remaining transitions and population variables whose contribution to evolution of the target populations is insignificant. This is achieved by defining appropriate coupling coefficients

as a measure of the influence of the transitions and population variables on the target populations.

3.1. Direct Coupling Coefficients

Transitions and population variables are coupled through direct and indirect influence on each other. The direct coupling coefficients are defined as a measure of the direct influence of a transition on the evolution of a population variable, or the other way around. The direct influence of a transition on a population variable is measured differently to the direct influence of a population variable on a transition. Thus, their definitions are also given separately.

Definition 1. *Given an arbitrary transition τ_j and a population variable x_i, the direct coupling coefficient of τ_j on x_i is defined as:*

$$c_{x_i,\tau_j} = \frac{|d^i_{\tau_j}\ N_{\tau_j}|}{\sum_{\tau\in\mathcal{T}} |d^i_\tau\ N_\tau|} \tag{5}$$

where d^i_τ is the update of x_i caused by the firing of transition τ, N_τ is the firing count of transition τ during a simulation run.

Intuitively, c_{x_i,τ_j} measures the proportional contribution of the transition τ_j to the evolution of population variable x_i during a simulation run. With smaller values of c_{x_i,τ_j}, the removal of transition τ_j from simulation will be less likely to immediately induce a significant impact on the evolution of population variable x_i.

Definition 2. *Given an arbitrary transition τ_j and a population variable x_i, the direct coupling coefficient of x_i on τ_j is defined as:*

$$c_{\tau_j,x_i} = \begin{cases} 1, & \textit{if population variable } x_i \textit{ contributes to transition } \tau_j \\ 0, & \textit{otherwise} \end{cases} \tag{6}$$

where we say x_i contributes to τ_j if and only if agent S_i appears at the reactant side of τ_j (assuming expressing transitions in the chemical reaction style) or the rate of τ_j depends on x_i.

Since removing a population variable which contributes to a transition will immediately invalidate the transition, the direct coupling coefficient of a population variable on a transition is either 1 or 0.

The direct coupling coefficients between two population variables or two transitions are always defined to be zero because we assume they are never directly coupled:

$$c_{x_i,x_j} = 0, \quad \forall(x_i, x_j)$$
$$c_{\tau_i,\tau_j} = 0, \quad \forall(\tau_i, \tau_j)$$

3.2. Evaluating the Firing Count of Transitions

A key point for the computation of direct coupling coefficients is the evaluation of N_τ, the firing count of transitions during a simulation run (all other factors in the definitions of direct coupling coefficients in Equations 5 and 6 can be directly obtained from the PCTMC description). The most straightforward way to evaluate N_τ is to simulate the PCTMC for a few experimental runs, and use the average firing count of each transition over those experimental simulation runs for the computation of direct coupling coefficients. However, the direct coupling coefficients computed in this way can be biased to the experimental simulation runs. Thus, in order to avoid the bias, we compute the expected firing count of transitions over infinite simulation runs through a deterministic model. This is achieved by moment approximation of a PCTMC with additional dummy population variables representing the counter of the firing of transitions. Specifically, based on a PCTMC $\mathcal{P} = \Big(\mathbf{x} = (x_1, \ldots, x_n), \mathcal{T} = (\tau_1, \ldots, \tau_m), \mathbf{x}_0 = \big(x_1(0), \ldots, x_n(0)\big)\Big)$ for stochastic simulation, we can construct another PCTMC $\mathcal{P}' = (\mathbf{x}', \mathcal{T}', \mathbf{x}'_0)$, in which:

- $\mathbf{x}' = (x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m})$, where x_{n+i} $(1 \leq i \leq m)$ is a dummy population variable representing the counter of the firing of transition τ_i.
- $\mathcal{T}' = (\tau'_1, \ldots, \tau'_m)$, where for $1 \leq i \leq m$, $\tau'_i = (r_{\tau'_i}(\mathbf{x}'), \mathbf{d}_{\tau'_i})$ such that

$r_{\tau'_i}(\mathbf{x}') = r_{\tau_i}(\mathbf{x})$, $\mathbf{d}_{\tau'_i} = (d^1_{\tau'_i}, \ldots, d^n_{\tau'_i}, d^{n+1}_{\tau'_i}, \ldots, , d^{n+m}_{\tau'_i})$ in which

$$d^j_{\tau'_i} = \begin{cases} d^j_{\tau_i} & \text{if } 1 \leq j \leq n \\ 0 & \text{if } n+1 \leq j \leq n+m \wedge j \neq n+i \\ 1 & \text{if } j = n+i \end{cases}$$

- $\mathbf{x}'_0 = \big(x'_1(0), \ldots, x'_n(0), x'_{n+1}(0), \ldots, x'_{n+m}(0)\big)$ where

$$x'_i(0) = \begin{cases} x_i(0) & \text{if } 1 \leq i \leq n \\ 0 & \text{if } n+1 \leq i \leq n+m \end{cases}$$

Intuitively, the above PCTMC will increase the counter for a transition by one whenever the transition is fired. Then, by computing the first moments of the population variables using moment approximation as described in Section 2.2, we can evaluate $N_{\tau_i} = \mathbb{E}[x_{n+i}](t_e)$ for $1 \leq i \leq m$, where t_e is the end time of the simulation. Note that if all the transition rates of the PCTMC are constants or linear functions of population variables, then we only need to compute the first moments of population variables by numerically solving Equation (2). Otherwise, we must obtain the ODEs for the first and second order moments according to Equations (2,3,4), and apply moment-closure methods to close the system at the second order (applying moment-closure methods at the first order usually does not work well). Furthermore, we can also apply moment ODE reduction methods to significantly reduce the number of ODEs for the joint moments which can tremendously reduce the cost of moment approximation [37]. Since we only need to compute the moments up to the first order or the second order of the PCTMC, the computational cost of moment approximation is typically fairly low compared with the total stochastic simulation cost.

3.3. Directed Coupling Graph

With the evaluation of direct coupling coefficients, we can construct a directed coupling graph (DCG) for an arbitrary PCTMC. The definition of DCG is given as follows:

Definition 3. *The directed coupling graph for a PCTMC with n population variables and m transitions is a graph consisting of $m + n$ nodes, in which each node represents a population variable or a transition in the PCTMC, and there exists a weighted directed edge from node i to node j if the direct coupling coefficient $c_{i,j} > 0$. In this case, $c_{i,j}$ is the weight for the edge.*

A point to note is that the arrow direction in the DCG is from the impacted node to the influencing node. This is due to its convenience for the calculation of coupling coefficients (which will be described later) starting from the nodes representing the target populations following the arrow directions. For example, consider a PCTMC which consists of five population variables $(x_1, x_2, x_3, x_4, x_5)$ for agent types S_1, S_2, S_3, S_4, S_5 respectively, and five transitions $(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5)$ as follows:

$$\begin{aligned}
S_1 &\longrightarrow_{\tau_1} S_4 && \text{at rate } \alpha_1 \times x_1 \\
S_1 + S_2 &\longrightarrow_{\tau_2} 2S_1 && \text{at rate } \alpha_2 \times x_1 \times x_2 \\
S_2 + S_3 &\longrightarrow_{\tau_3} 2S_3 && \text{at rate } \alpha_3 \times x_2 \times x_3 \\
S_3 &\longrightarrow_{\tau_4} S_4 && \text{at rate } \alpha_4 \times x_3 \\
S_4 &\longrightarrow_{\tau_5} S_5 && \text{at rate } \alpha_5 \times x_4
\end{aligned}$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$ are five constants. Assuming $N_{\tau_1} = 70$, $N_{\tau_2} = 30$, $N_{\tau_3} = 10$, $N_{\tau_4} = 30$, $N_{\tau_5} = 100$, then the corresponding DCG is depicted in Figure 1. As can be seen from the graph, there is an edge from node x_1 to node τ_1 since $c_{x_1,\tau_1} = N_{\tau_1}/(N_{\tau_1} + N_{\tau_2}) = 0.7$, an edge from node τ_1 to node x_1 since x_1 contributes to τ_1 (S_1 appears in the reactant side of τ_1 and the rate function of τ_1 depends on x_1), thus $c_{\tau_1,x_1} = 1$. There is no edge from node τ_1 to node x_4 because $c_{\tau_1,x_4} = 0$ since x_4 makes no direct contribution to τ_1 (S_4 does not appear on the reactant side of τ_1 nor does the rate function of τ_1 depends on x_4).

In the above example, if x_1 is a target population, then removing transitions τ_3, τ_4 and τ_5 will not induce a **direct** impact on the evolution of x_1, since $c_{x_1,\tau_3} = c_{x_1,\tau_4} = c_{x_1,\tau_5} = 0$. But, from the model definition, we can clearly

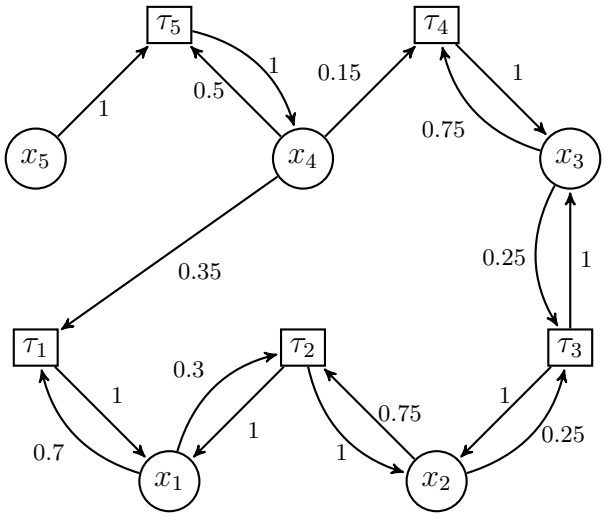

Figure 1: The directed coupling graph for the example PCTMC

see that removing τ_5 will make zero impact on x_1 either directly or indirectly, whereas removing τ_3 and τ_4 can affect x_1 through an indirect coupling with population variable x_2 and x_3. This indirect coupling effect can be captured by a propagation method based on the DCG.

3.4. Coupling Propagation

Transitions can influence the evolution of population variables by coupling propagation through intermediate populations and transitions. Moreover, the further away from the target population a transition is, the smaller the effect of changing or removing this transition should be. Thus, for a target population x_t and a transition τ which are not directly connected in the DCG, we quantify the **indirect coupling coefficient** of the transition τ to the target population x_t by a path dependent coefficient $c^{\gamma}_{x_t,\tau}$, which is the product of the direct coupling coefficients along an acyclic path γ from node x_t to node τ in the DCG:

$$c^{\gamma}_{x_t,\tau} = \prod_{ij\in\gamma} c_{i,j}.$$

Intuitively, $c^{\gamma}_{x_t,\tau}$ measures the estimated influence of transition τ on the evolution of the target population x_t through the coupling propagated along the path

γ. Take the DCG in Figure 1 as an example, suppose x_1 is the target population, then the influence of τ_3 on the evolution of x_1 has to propagate through its direct influence on x_2, x_2's influence on τ_2, and finally reach x_1 through the influence of τ_2. Doing so, this indirect influence is estimated by the product of the direct coupling coefficients along the path x_1, τ_2, x_2, τ_3 to capture this diminishing propagation effect, such that $c^{\gamma}_{x_1,\tau_3} = c_{x_1,\tau_2} \times c_{\tau_2,x_2} \times c_{x_2,\tau_3}$, where $\gamma = \{x_1\tau_2, \tau_2 x_2, x_2\tau_3\}$.

Furthermore, for the purpose of model decoupling, we assume all coupling (both direct and indirect) with respect to the target populations which are less than a specific threshold ϵ can be ignored. This means that if the maximum of path dependent coefficients over all possible paths from a target population to a transition is less than ϵ, then all the coupling of that transition to the target population can be ignored. Therefore, if we characterize the influence of an arbitrary transition τ on the evolution of a target population x_t by a coupling coefficient $C_{x_t,\tau}$, which is defined as follows:

Definition 4. *Given an arbitrary transition τ and a target population variable x_t, the **coupling coefficient** of τ on x_t is defined as the maximum of the path dependent coefficients from node x_t to node τ in the DCG:*

$$C_{x_t,\tau} = \begin{cases} \max\limits_{all\ paths\ \gamma} c^{\gamma}_{x_t,\tau}, & \text{if there exists a path from } x_t \text{ to } \tau \text{ in the DCG} \\ 0, & \text{otherwise} \end{cases}$$

Then, we can specify whether a transition is coupled with a target population by simply checking if $C_{x_t,\tau} > \epsilon$. Take the DCG in Figure 1 as an example, again if x_1 is a target population, then we can obtain $C_{x_1,\tau_1} = 0.7$, $C_{x_1,\tau_2} = 0.3$, $C_{x_1,\tau_3} = 0.3 \times 1 \times 0.25 = 0.075$, $C_{x_1,\tau_4} = 0.3 \times 1 \times 0.25 \times 1 \times 0.75 = 0.05625$, $C_{x_1,\tau_5} = 0$. Therefore, if we set $\epsilon = 0.01$, only τ_5 can be decoupled. However, if we set $\epsilon = 0.1$, then τ_3 and τ_4 will also be decoupled. Thus, the decoupling threshold ϵ can be thought of as a parameter to control the extent of model decoupling.

Apart from the above intuition, there is an important point to note that the coupling coefficients, which are characterized by the maximum of the path

dependent coefficients, have the nice property that if $C_{x_t,\tau} > \epsilon$ and $C_{x_t,\tau} = c^{\gamma}_{x_t,\tau}$, then for a node v which is on the path γ, it is certain that $C_{x_t,v} > \epsilon$. Thus node v will definitely not be decoupled. As a result, for all transitions which have a significant path dependent coefficient from the target population, the transitions and populations along this path will certainly be captured in the pivotal part of the decoupled PCTMC as derived subsequently. This guarantees that the coupling coefficient of any transition to the target population will not be changed in the decoupled PCTMC. In contrast, this property would not be preserved if we were to use other metrics, e.g., the sum of path dependent coefficients, to characterize coupling coefficients.

Furthermore, given the DCG of a PCTMC and a set of target populations, we can easily calculate the coupling coefficients of all the transitions on these target populations by a modified version of Dijkstra's algorithm [38]. Dijkstra's algorithm was originally introduced by Dijkstra, and many variants have been proposed to enhance its efficiency for calculating the shortest paths from a single source node to all other nodes in a graph [39]. Calculating coupling coefficients is a shortest path problem where the "shortest" path is that with the maximum product of direct coupling coefficients representing edge weights. This is the only modification needed to apply Dijkstra's algorithm to our problem. The detail of our modified version of Dijkstra's algorithm for calculating coupling coefficients is given in Algorithm 2, where we use a max-priority queue [39] to efficiently search nodes with maximum path dependent coefficients. Let M be the number of target populations, V and E be the number of nodes and edges in the DCG, respectively, the time complexity of the algorithm is $O(M(V + E) \log V)$.

3.5. The Decoupling Algorithm

Given an arbitrary PCTMC $\mathcal{P} = (\mathbf{x}, \mathcal{T}, \mathbf{x}_0)$, a set of target populations $\mathbf{x}_t$, and a decoupling threshold ϵ, the decoupling algorithm splits the PCTMC into the pivotal part $\hat{\mathcal{P}} = (\hat{\mathbf{x}}, \hat{\mathcal{T}})$ and the trivial part $\check{\mathcal{P}} = (\check{\mathbf{x}}, \check{\mathcal{T}})$. Specifically, transitions whose coupling coefficients with respect to any target population variable is larger than ϵ are classified as belonging to the pivotal part. Population vari-

Algorithm 2 Algorithm for Calculating Coupling Coefficients

Require: The DCG G, target population variables $\mathbf{x}_t$

for all x_t in $\mathbf{x}_t$ **do**
  $C_{x_t,x_t} \leftarrow 1$ {initialize the coupling coefficient of a target population on itself to the maximum value 1}
  Create an empty max-priority queue Q
  for all node v in G **do**
    if $v \neq x_t$ **then**
      $C_{x_t,v} \leftarrow 0$ {initialize the coupling coefficient of all other nodes on the target population to the minimum value 0}
    end if
    Q.add(v, $C_{x_t,v}$) {add node v to Q with priority $C_{x_t,v}$}
  end for
  while Q is not empty **do**
    $u \leftarrow$ Q.extract_max() {remove and return node with maximum priority}
    for all neighbour v of u {v is a neighbour of u iff $c_{u,v} > 0$} **do**
      if v is still in Q **then**
        $C_{tmp} \leftarrow C_{x_t,u} \times c_{u,v}$
        if $C_{tmp} > C_{x_t,v}$ **then**
          $C_{x_t,v} \leftarrow C_{tmp}$
          Q.update_priority(v, $C_{x_t,v}$) {update node v's priority to $C_{x_t,v}$}
        end if
      end if
    end for
  end while
end for
return $C_{x_t,\tau} \quad \forall x_t \in \mathbf{x}_t, \tau \in \mathcal{T}$

ables which directly contribute to any transitions in the pivotal part also belong to the pivotal part. The remaining transitions and population variables are decoupled, and so belong to the trivial part. The whole algorithm is summarised in Algorithm 3.

4. The Reduction Method

In this section, we present our model reduction method based on the decoupled representation of a PCTMC. Specifically, transitions and population variables in the pivotal part must be preserved since their contributions to the evolution of target populations are significant. Therefore, our strategy is to reduce the simulation cost for the trivial part of the decoupled PCTMC but without causing significant deviation on the evolution of the target populations. This is achieved by firstly truncating the coupling coefficients of the population variables and transitions in the trivial part with respect to the target populations.

4.1. Coupling Truncation

First of all, let us call any transitions in the trivial part which can directly influence any population variables in the pivotal part of the decoupled PCTMC, *border transitions*. Specifically, the set of border transitions are defined as follows:

Definition 5. *Let $\mathcal{T}_b$ denote the set of* ***border transitions****, then for any transition $\tau \in \check{\mathcal{T}}$, we let $\tau \in \mathcal{T}_b \iff \exists x_i$ such that $x_i \in \hat{\mathbf{x}} \wedge c_{x_i,\tau} > 0$.*

More intuitively speaking, these border transitions are the border nodes of the trivial part of the PCTMC with respect to the pivotal part of the PCTMC in the DCG. More importantly, they are the only links through which the transitions and population variables in the trivial part can affect the evolution of target populations, either directly and indirectly. Therefore, we make these border transitions independent of any population variables in the trivial part of the decoupled PCTMC, whilst retaining their influence on target populations.

Algorithm 3 Model Decoupling Algorithm

Require: The original PCTMC $\mathcal{P} = (\mathbf{x}, \mathcal{T}, \mathbf{x}_0)$, target population variables $\mathbf{x}_t$, decoupling threshold ϵ

Define $\hat{\mathbf{x}} = \emptyset$, $\hat{\mathcal{T}} = \emptyset$, $\check{\mathbf{x}} = \emptyset$, $\check{\mathcal{T}} = \emptyset$
for all τ in the transition set $\mathcal{T}$ of $\mathcal{P}$ **do**
    $\beta_\tau \leftarrow 0$ {indicate whether τ can be decoupled}
    for all x_t in $\mathbf{x}_t$ **do**
        if $C_{x_t,\tau} > \epsilon$ **then**
            $\beta_\tau \leftarrow 1$
            break
        end if
    end for
    if $\beta_\tau = 1$ **then**
        add τ to $\hat{\mathcal{T}}$
    else
        add τ to $\check{\mathcal{T}}$
    end if
end for
for all x_i in the population vector $\mathbf{x}$ of $\mathcal{P}$ **do**
    $\beta_i \leftarrow 0$ {indicate whether x_i can be decoupled}
    for all τ in $\hat{\mathcal{T}}$ **do**
        if x_i contributes to τ directly **then**
            $\beta_i \leftarrow 1$
            break
        end if
    end for
    if $\beta_i = 1$ **then**
        add x_i to $\hat{\mathbf{x}}$
    else
        add x_i to $\check{\mathbf{x}}$
    end if
end for
return The pivotal part $\hat{\mathcal{P}} = (\hat{\mathbf{x}}, \hat{\mathcal{T}})$, the trivial part $\check{\mathcal{P}} = (\check{\mathbf{x}}, \check{\mathcal{T}})$

Subsequently all the couplings of population variables and other transitions in the trivial part with the target populations are truncated, meaning that they no longer impact the evolution of target populations, and so can be removed from the simulation without causing any deviation in the evolution of target populations.

Coupling truncation is achieved by transforming the border transitions using the following approach: let $\tau = (r_\tau(\mathbf{x}), \mathbf{d}_\tau) \in \mathcal{T}_b$ be an arbitrary border transition, and n be the number of population variables in $\mathbf{x}$. Then for all $i \in [1, 2, \ldots, n]$, we set $d_\tau^i = 0$ if $x_i \in \check{\mathbf{x}}$. Moreover, we approximate $r_\tau(\mathbf{x})$ by a constant value N_τ/t_e if there is any population variable in the trivial part that contributes to the transition directly. After the above transformation, the border transition will not involve any population variables in the trivial part but its influence on the pivotal population variables are approximately preserved. Moreover, since the influence of the border transitions on the evolution of target populations are rather limited (their coupling coefficients on any target population must be less than ϵ), the approximation of their rates will be unlikely to cause significant deviation on the target populations.

4.2. Reduction Algorithm

We summarise the reduction process in Algorithm 4 in which steps 2 to 6 derive border transitions; Steps 7 to 12 truncate coupling of population variables and transitions in the trivial part by transforming border transitions. Steps 13 to 16 aggregate transformed border transitions with the same update vectors. Steps 17 to 19 remove any update of population variables in the trivial part caused by transitions in the pivotal part to zero (these population variables do not contribute to the transitions directly but can nevertheless appear in the production side of the transitions).

Algorithm 4 Model Reduction Algorithm

Require: The pivotal part $\hat{\mathcal{P}} = (\hat{\mathbf{x}}, \hat{\mathcal{T}})$, the trivial part $\check{\mathcal{P}} = (\check{\mathbf{x}}, \check{\mathcal{T}})$

Define $\mathcal{T}_b = \emptyset$
for all τ in $\check{\mathcal{T}}$ **do**
  if $\exists x_i$ such that $x_i \in \hat{\mathbf{x}} \wedge c_{x_i,\tau} > 0$ **then**
    Add τ to $\mathcal{T}_b$
  end if
end for
for all τ in $\mathcal{T}_b$ **do**
  if $\exists x_i$ such that $x_i \in \check{\mathbf{x}} \wedge c_{\tau,x_i} = 1$ **then**
    Set $r_\tau = N_\tau / t_e$
  end if
  $\forall i$, set $d_\tau^i = 0$ if $x_i \in \check{\mathbf{x}}$
end for
while $\exists\ \tau_i \in \mathcal{T}_b$, $\tau_j \in \mathcal{T}_b$ such that $i \neq j \wedge \mathbf{d}_{\tau_i} = \mathbf{d}_{\tau_j} \wedge r_{\tau_i} \in \mathbb{R} \wedge r_{\tau_j} \in \mathbb{R}$ **do**
  Create a new τ, set $\mathbf{d}_\tau = \mathbf{d}_{\tau_i}$ and $r_\tau = r_{\tau_i} + r_{\tau_j}$
  Remove τ_i and τ_j from $\mathcal{T}_b$, and add τ to $\mathcal{T}_b$
end while
for all τ in $\hat{\mathcal{T}}$ **do**
  $\forall i$, set $d_\tau^i = 0$ if $x_i \in \check{\mathbf{x}}$
end for
Set $\hat{\mathcal{T}} = \hat{\mathcal{T}} \cup \mathcal{T}_b$
return The reduced PCTMC $\hat{\mathcal{P}} = (\hat{\mathbf{x}}, \hat{\mathcal{T}}, \hat{\mathbf{x}}_0)$ for simulation where $\hat{\mathbf{x}}_0$ is a subvector of $\mathbf{x}_0$ which gives the corresponding initial value for $\hat{\mathbf{x}}$

After the reduction algorithm is applied, only transitions and population variables in the pivotal part, together with the aggregated border transitions, are retained; all the other population variables and transitions are discarded. As an illustration, for the example PCTMC in Section 3.3, let x_1 be the target population, $\epsilon = 0.1$, then the pivotal part of the decoupled PCTMC consists of $\{x_1, x_2, \tau_1, \tau_2\}$, the trivial part consists of $\{x_3, x_4, x_5, \tau_3, \tau_4, \tau_5\}$ where τ_3 is a

border transition because $c_{x_2,\tau_3} = 0.25$. After the model reduction algorithm is applied, the PCTMC only consists of $\{x_1, x_2, \tau_1, \tau_2, \tau_3\}$ where the transitions are transformed as follows:

$$\begin{aligned} S_1 &\longrightarrow_{\tau_1} \emptyset \quad \text{at rate } \alpha_1 \times x_1 \\ S_1 + S_2 &\longrightarrow_{\tau_2} 2S_1 \quad \text{at rate } \alpha_2 \times x_1 \times x_2 \\ S_2 &\longrightarrow_{\tau_3} \emptyset \quad \text{at rate } N_{\tau_3}/t_e \end{aligned}$$

Here the only approximation we make which will impact the evolution of x_1 is the transition rate of τ_3. However, since the contribution of τ_3 to the evolution of x_1 is minor, the approximation is unlikely to cause significant deviation in the dynamics of x_1.

5. Reduced Model Validation

We also note that the extent of the reduction can vary significantly when choosing different values of the decoupling threshold parameter ϵ. Specifically, with a larger value of ϵ, more transitions and population variables will be removed from the stochastic simulation. However, a larger deviation in the evolution of target populations is also expected. On the other hand, when ϵ tends to zero, the deviation caused by the approximation of the rate of border transitions tends to be smaller, thus the dynamics of target populations in the reduced model will converge to be the same as in the full model. Thus, we expect that if $\mathbf{x}_t$ is the set of target populations, and ϵ is the decoupling threshold, then for $0 < t < t_e$:

$$\lim_{\epsilon \to 0} \mathrm{D}\big(\mathrm{P}(\mathbf{x}_t, t \mid \mathcal{P}), \mathrm{P}(\mathbf{x}_t, t \mid \hat{\mathcal{P}})\big) = 0$$

where $\mathrm{P}(\mathbf{x}_t, t \mid \mathcal{P})$ and $\mathrm{P}(\mathbf{x}_t, t \mid \hat{\mathcal{P}})$ are the probability distributions of the target populations at time t given the full model and the reduced model, respectively; $\mathrm{D} : \mathrm{P} \times \mathrm{P} \to \mathbb{R}^+$ is a suitable statistical distance between the two probability distributions. Accordingly, in general, when we choose a small value for ϵ, the model will behave well. However, there is still no direct estimate of the amount of error which will be caused by the reduction. Therefore, we also propose an

efficient validation method to estimate the error caused by model reduction. Specifically, instead of computing the statistical distance between the probability distributions of the target populations in the full and reduced models, we estimate the error on the mean dynamics of target populations which is more computationally efficient to evaluate. Again, this is achieved by computing the first moment of target populations using the moment approximation method. Note that the first moment of target populations of the full model can be obtained while evaluating the firing count of transitions for the computation of direct coupling coefficients. Thus, we only need to compute the first moment of the target populations in the reduced model for validation.

Concretely, after obtaining the first moment of the target populations through moment approximation in both models, we can evaluate the error caused by model reduction on an arbitrary target population x_t at time t by:

$$err(x_t, t) = \frac{\mid \mathbb{E}[x_t, t \mid \mathcal{P}] - \mathbb{E}[x_t, t \mid \hat{\mathcal{P}}] \mid}{\mathbb{E}[x_t, t \mid \mathcal{P}]} \tag{7}$$

where $\mathbb{E}[x_t, t \mid \mathcal{P}]$ and $\mathbb{E}[x_t, t \mid \hat{\mathcal{P}}]$ are the first moment of the target population x_t at time t in the full model and the reduced model, respectively; $err(x_t, t)$ is the validation error of the target population x_t at time t.

Overall, validation errors increase monotonically with ϵ. Therefore, we can choose a maximal value of ϵ below a validation error to achieve the largest level of reduction, but still retain acceptable accuracy on target populations.

6. Case Studies

In this section, we apply our technique to two case studies: a multi-class version of the well-known SIR (Susceptible-Infected-Recovered) model for disease spread and a bike-sharing model whose parameters are fitted with historical data from Santander Cycles, the public bike-sharing system in London. In the experiments for both case studies, we run Gillespie's SSA on the full models and the reduced models with different values of ϵ. The applicability and power of our technique is evaluated by the size of the reduced model, the decrease in

simulation time, and the statistical accuracy on the target populations. In order to allow a fair statistical accuracy evaluation, we also require enough simulation runs to allow the simulation sample means to converge. More specifically, we require the width of the 95% confidence interval of the simulation sample mean to be less than 1% of the value of the simulation sample mean at any point in time. All the experiments were run on a Linux machine with 16 GB memory size and 3.4 GHz Intel core i7 CPU.

6.1. The Multi-class SIR Model

We first consider the well-known SIR model. The SIR model studies the spread of disease by considering the dynamics of three types of individuals who are susceptible (S), infected (I) and recovered (R) with respect to the disease. Susceptible individuals are those who have not contracted the disease; they may get infected by an encounter with another individual who currently carries the disease; finally, infected individuals recover after some time. The model can be studied by a PCTMC containing the following transitions:

$$\begin{aligned} S &\rightarrow I \quad \text{at } \beta \,\#(S)\,\#(I) \\ I &\rightarrow R \quad \text{at } \gamma \,\#(I) \end{aligned}$$

where β and γ are positive reals describing the infection rate and the recovery rate, respectively; $\#(S)$ and $\#(I)$ denote the population of suspectibles and infected individuals, respectively. Here we consider a multi-class variant with N classes of individuals with class-specific infection rates and recovery rates [40, 41]. Specifically, the transitions for the PCTMC of the multi-class SIR model are extended as follows:

$$\begin{aligned} S_i + I_j \rightarrow I_i + I_j \quad & \text{at } \beta_{i,j}\,\#(S_i)\,\#(I_j) \quad && \forall i, j \in [1, N] \\ I_i \rightarrow R_i \quad & \text{at } \gamma_i\,\#(I_i) && \forall i \in [1, N] \end{aligned}$$

where S_i, I_i, R_i represent a susceptible, infected and recovered individual of class i; $\beta_{i,j}$ denotes the rate for a susceptible individual of class i to be infected by an infected individual of class j.

6.1.1. Experiment Setup

In our experiments, we consider a multi-class SIR model with $N = 30$. Furthermore, experiments are carried out on 30 randomly chosen different instantiations of the SIR model in order to avoid the bias caused by a particular set of parameters. Specifically, in each instantiation, $\beta_{i,j}$ and r_i are set to different values sampled from a uniform distribution between $[0, 1]$, the initial population for susceptibles of each class is set to be a different integer sampled from a uniform distribution between $[0, 50]$, the initial infected population of each class is set to be a random value generated from a uniform distribution between $[0, 5]$, the recovered populations of all classes are set to zero.

For each model instantiation, suppose we are interested in how many individuals of a randomly chosen class are infected by the disease before the disease is extinguished, thus the population of recovered individuals of that class is set as the target population. Then, we conduct stochastic simulation for both the full/original model and the reduced model after applying our reduction method. The final time of a simulation run is set to 10, which allows the disease to be extinguished. The number of stochastic simulation runs for each model instantiation is set to 1000 for both the full and reduced models which allows the sample means to converge. Furthermore, in each experiment with respect to a model instantiation, we set ϵ to different values from 0.001 to 0.1 to see the performance of our reduction method with different decoupling thresholds.

6.1.2. Evaluation Metrics

To evaluate the level of reduction by our method, we outline the average number of populations and transitions, the time cost per simulation run for the reduced models compared with their counterparts for the full models as well as the time cost of the reduction process (the time costs of model decoupling, reduction and validation are all included). The accuracy of the reduced models are evaluated by four metrics: the validation error on the target populations using moment approximation, the relative error on the mean and standard deviation of target populations in the stochastic simulation runs of the reduced

models compared with the corresponding values in the simulation runs of the full models, the Bhattacharyya distance of the probability distributions of the target populations in the stochastic simulation runs of the reduced models and their corresponding full models. Specifically, all the four metrics are calculated for 200 time points evenly distributed in the simulation time duration for each experiment. The relative error in the mean and standard deviation at each time point are calculated by the distance between their values in the reduced and full models divided by the value in the full model which is similar to Equation 7, such that:

$$err_f(x_t, t) = \frac{\mid f(x_t, t \mid \mathcal{P}) - f(x_t, t \mid \hat{\mathcal{P}}) \mid}{f(x_t, t \mid \mathcal{P})}$$

where f denotes the mean or std function, $f(x_t, t \mid \mathcal{P})$ and $f(x_t, t \mid \hat{\mathcal{P}})$ denote the mean or the standard deviation of the target population x_t at time t in the simulation of the full model and the reduced model, respectively. The Bhattacharyya distance is a statistical distance which measures the similarity of two probability distributions [42]. Concretely, let $\mathcal{D}_t$ be the value domain of a target population x_t, then the Bhattacharyya distance of the probability distributions of x_t in the full and reduced models at time t is defined as:

$$\mathrm{D}_B(x_t, t) = -\ln \sum_{z \in \mathcal{D}_t} \sqrt{\mathrm{P}(x_t = z, t \mid \mathcal{P}) \times \mathrm{P}(x_t = z, t \mid \hat{\mathcal{P}})} \tag{8}$$

where $0 \leq \mathrm{D}_B(x_t, t) \leq \infty$, and $\mathrm{D}_B(x_t, t) = 0$ if $\mathrm{P}(x_t, t \mid \mathcal{P})$ and $\mathrm{P}(x_t, t \mid \hat{\mathcal{P}})$ perfectly overlap. Lastly, we call $err(x_t, t)$ (the validation error), $err_f(x_t, t)$ and $\mathrm{D}_B(x_t, t)$ error metrics. Thus since there is only one target population in each experiment of this case study, each experiment will generate 200 error samples along the simulation time duration for each error metric.

6.1.3. Experiment Results

We give the average number of populations and transitions, average time cost per simulation run with 95% confidence interval with different values of ϵ on the 30 instantiations of the multi-class SIR model in Table 1. The time cost of the reduction process for all scenarios is similar, which is 1428 $\pm$ 102ms

	ϵ	Number of populations	Number of transitions	time cost per run (ms)
Full model	N/A	90	930	245 ± 8
Reduced model	1×10^{-3}	43 ± 1.1	883 ± 20	113 ± 8
	2.5×10^{-3}	30 ± 1.5	745 ± 54	62 ± 7
	5×10^{-3}	25 ± 0.5	558 ± 24	40 ± 4
	7.5×10^{-3}	23 ± 0.9	448 ± 40	30 ± 5
	1×10^{-2}	21 ± 1.1	384 ± 42	23 ± 4
	2.5×10^{-2}	11 ± 1.4	93 ± 28	4 ± 1
	5×10^{-2}	4 ± 0.9	12 ± 5	0.7 ± 0.2
	7.5×10^{-2}	3 ± 0.6	3 ± 2.1	0.3 ± 0.1
	1×10^{-1}	2 ± 0	2 ± 0	0.2 ± 0.03

Table 1: Reduction metrics of our method on the multi-class SIR model case study

(millisecond) on average with 95% confidence interval. Table 2 gives the corresponding average (also with 95% confidence interval) for the error metrics over all the error samples generated in the experiments on the 30 instantiations of the multi-class SIR model.

It can be seen that our method can significantly reduce the number of population variables and transitions as well as the simulation time cost even with a small decoupling threshold. With larger thresholds, more transitions are removed and thus the simulation time is further reduced. The overhead of the reduction process for the multi-class SIR model is approximately the time cost of 6 simulation runs of the full model. This means the overhead cost of the reduction process is almost negligible if a large number of simulation runs is required for checking the statistical properties of the target populations. From Table 2, we can see that the error caused by model reduction on the target populations is well approximated by the validation error. We observe that the deviation of the statistical properties on the target population is rather low when $\epsilon \leq 2.5 \times 10^{-2}$. However, we find that setting $\epsilon > 2.5 \times 10^{-2}$ will cause significant error. This means that when $\epsilon \leq 2.5 \times 10^{-2}$, the transitions and population variables we

ϵ	Validation error (err)	Simulation error on mean (err_{mean})	Simulation error on std (err_{std})	Bhattacharyya distance (D_B)
1×10^{-3}	$5.2 \pm 0.5 \times 10^{-3}$	$5.8 \pm 0.4 \times 10^{-3}$	$3.2 \pm 0.2 \times 10^{-2}$	$8.7 \pm 1.1 \times 10^{-4}$
2.5×10^{-3}	$6.2 \pm 0.6 \times 10^{-3}$	$6.4 \pm 0.5 \times 10^{-3}$	$3.4 \pm 0.2 \times 10^{-2}$	$1.1 \pm 0.1 \times 10^{-3}$
5×10^{-3}	$8.4 \pm 1.0 \times 10^{-3}$	$8.3 \pm 0.7 \times 10^{-3}$	$3.6 \pm 0.2 \times 10^{-2}$	$1.2 \pm 0.1 \times 10^{-3}$
7.5×10^{-3}	$8.7 \pm 1.1 \times 10^{-3}$	$8.8 \pm 0.7 \times 10^{-3}$	$3.6 \pm 0.2 \times 10^{-2}$	$1.3 \pm 0.3 \times 10^{-3}$
1×10^{-2}	$1.0 \pm 0.1 \times 10^{-2}$	$8.9 \pm 0.8 \times 10^{-3}$	$3.9 \pm 0.2 \times 10^{-2}$	$1.5 \pm 0.2 \times 10^{-3}$
2.5×10^{-2}	$1.4 \pm 0.2 \times 10^{-2}$	$1.4 \pm 0.1 \times 10^{-2}$	$4.0 \pm 0.2 \times 10^{-2}$	$1.6 \pm 0.1 \times 10^{-3}$
5×10^{-2}	$1.3 \pm 0.1 \times 10^{-1}$	$1.2 \pm 0.1 \times 10^{-1}$	$1.7 \pm 0.1 \times 10^{-1}$	1.1 ± 0.2
7.5×10^{-2}	$5.0 \pm 0.2 \times 10^{-1}$	$5.4 \pm 0.1 \times 10^{-1}$	$4.8 \pm 0.2 \times 10^{-1}$	3.6 ± 0.2
1×10^{-1}	$6.2 \pm 0.1 \times 10^{-1}$	$6.0 \pm 0.1 \times 10^{-1}$	$5.1 \pm 0.1 \times 10^{-1}$	3.6 ± 0.2

Table 2: Validation and simulation errors of our reduction method on the multi-class SIR model case study

removed or abstracted all have minor impact on the evolution of target populations. In contrast, if ϵ is set to be larger than 2.5×10^{-2}, the model will no longer retain enough of the original transitions to remain a faithful representation of the full model: transitions have been removed or abstracted that do have a significant influence on the target population. We believe the point at which this happens will depend on the structure of the particular model, and our validation step can be used to find this critical point before stochastic simulation runs are conducted. In this case, by setting $\epsilon = 2.5 \times 10^{-2}$, we can simulate the model with a significantly lower computational cost (from 245ms per run to 4ms per run) whilst still achieving very high accuracy.

6.2. The Bike-sharing Model

The second example is a PCTMC which models a public bike-sharing system. Bike-sharing systems are becoming more and more important for urban transportation. In such systems, users arrive at a station, pick up a bike, use it for a while, and then return it to another station of their choice. Recently, PCTMCs have been used to model bike-sharing systems [5, 6, 7]. Here, we consider a model of N stations. We use a PCTMC containing the following

transitions to represent the model:

$$Bike_i \rightarrow Slot_i + Journey^i_j \quad \text{at } \lambda_i \, p^i_j \qquad \forall i, j \in [1, N] \wedge i \neq j$$

$$Journey^i_j + Slot_j \rightarrow Bike_j \quad \text{at } \#(Journey^i_j) \, \mu^i_j \qquad \forall i, j \in [1, N] \wedge i \neq j$$

where $Bike_i$ and $Slot_i$ denote an available bike or slot in station i, respectively; $Journey^i_j$ denotes a bike in transit from station i to station j. λ_i is the pickup rate of bikes in station i, p^i_j is the probability to choose station j as the destination of a trip when picking up a bike from station i. $1/\mu^i_j$ is the mean trip time from station i to station j. Note that since the focus of this paper is not the accuracy of bike availability prediction in stations, for simplicity, we assume journey durations are also exponentially distributed in the model. Although the assumption is generally not true in practice, it will not cause any problem on the demonstration of our model reduction method here.

6.2.1. Experiment Setup and Evaluation Metrics

We use the above PCTMC to model the journey dynamics from 8am to 9am in weekday mornings between 50 bike stations near Russell Square in central London. All the parameters in the model are calculated by journey data which is available online[1]. In total, 50 experiments are conducted for the bike-sharing model, where in each we choose the number of available bikes in one station and the number of available slots in another station as our target populations for model reduction. The number of stochastic simulation runs for the full model and the reduced model in each experiment is set to 10,000 in order to allow the simulation sample means to converge. Lastly, the same metrics as in the multi-class SIR model case study are used to evaluate the level of reduction and the error caused by reduction in this case study.

[1] `https://tfl.gov.uk/info-for/open-data-users/our-feeds?intcmp=3671#on-this-page-4`

6.2.2. Experiment Results

We give the reduction and error metrics on the bike-sharing model case study in Tables 3 and 4, respectively. The time cost of the reduction process in all the experiments is 3663 ± 141ms on average with 95% confidence interval. From the results, again, we observe that a large number of population variables and transitions can be removed, the time cost per simulation run can be tremendously reduced for the bike-sharing model even with a small decoupling threshold. The time cost of the reduction process is only about the cost of 20 simulation runs of the full model. Furthermore, we notice that even with $\epsilon = 1 \times 10^{-3}$, most of the transitions in the bike-sharing model can be removed or abstracted whilst still retaining a high simulation accuracy on the target populations. The time cost per simulation run can be reduced from 189ms to 6.8ms on average in this case. This indicates that in order to evaluate the probability distribution of bike and slot numbers in some particular stations, only some key journey dynamics need to be captured, other journey dynamics between stations in the bike-sharing model need not to be simulated explicitly. Increasing the value of ϵ will also increase the level of reduction, as well as the simulation error caused by reduction, steadily, and this trend (on the simulation error) is captured well by the validation error. This means that the modeller can decide the value of ϵ by the trade-off between level of reduction and error caused by reduction before stochastic simulation runs are conducted.

7. Conclusion

In this paper, we proposed an automatic model reduction method which can significantly accelerate stochastic simulation of PCTMCs assuming that only the statistical properties of a few target populations are to be checked. Our model reduction method exploits the coupling between population variables and transitions in the PCTMC. A decoupling threshold is used to control the extent of reduction. Population variables and transitions which have larger coupling coefficients than the decoupling threshold on the target populations are exactly

	ϵ	Number of populations	Number of transitions	time cost per run (ms)
Full model	N/A	914	1733	189 ± 27
Reduced model	1×10^{-3}	61 ± 6	85 ± 9	6.8 ± 0.9
	2.5×10^{-3}	54 ± 5	74 ± 7	5.9 ± 0.9
	5×10^{-3}	36 ± 4	55 ± 6	4.0 ± 0.5
	7.5×10^{-3}	27 ± 3	42 ± 5	3.0 ± 0.4
	1×10^{-2}	22 ± 2	34 ± 4	4.0 ± 0.3
	2.5×10^{-2}	10 ± 1	15 ± 2	0.9 ± 0.1
	5×10^{-2}	5.7 ± 0.6	7.4 ± 1.0	0.4 ± 0.06
	7.5×10^{-2}	4.4 ± 0.3	5.1 ± 0.6	0.3 ± 0.05
	1×10^{-1}	4.1 ± 0.3	4.1 ± 0.4	0.2 ± 0.03

Table 3: Reduction metrics of our method on the bike-sharing model case study

ϵ	Validation error (err)	Simulation error on mean (err_{mean})	Simulation error on std (err_{std})	Bhattacharyya distance (D_B)
1×10^{-3}	$1.1 \pm 0.4 \times 10^{-2}$	$1.0 \pm 0.3 \times 10^{-2}$	$1.2 \pm 0.3 \times 10^{-2}$	$1.6 \pm 0.1 \times 10^{-4}$
2.5×10^{-3}	$2.1 \pm 0.5 \times 10^{-2}$	$1.5 \pm 0.5 \times 10^{-2}$	$1.3 \pm 0.3 \times 10^{-2}$	$2.5 \pm 0.1 \times 10^{-4}$
5×10^{-3}	$2.5 \pm 0.8 \times 10^{-2}$	$2.3 \pm 0.8 \times 10^{-2}$	$2.0 \pm 0.6 \times 10^{-2}$	$7.8 \pm 0.5 \times 10^{-4}$
7.5×10^{-3}	$2.9 \pm 0.7 \times 10^{-2}$	$2.4 \pm 0.7 \times 10^{-2}$	$2.0 \pm 0.5 \times 10^{-2}$	$8.1 \pm 0.7 \times 10^{-4}$
1×10^{-2}	$2.9 \pm 0.7 \times 10^{-2}$	$2.6 \pm 0.7 \times 10^{-2}$	$2.5 \pm 0.6 \times 10^{-2}$	$1.1 \pm 0.5 \times 10^{-3}$
2.5×10^{-2}	$5.3 \pm 0.1 \times 10^{-2}$	$4.7 \pm 0.1 \times 10^{-2}$	$4.2 \pm 0.1 \times 10^{-2}$	$3.9 \pm 0.2 \times 10^{-3}$
5×10^{-2}	$5.7 \pm 0.1 \times 10^{-2}$	$5.4 \pm 0.1 \times 10^{-2}$	$5.3 \pm 0.1 \times 10^{-2}$	$5.8 \pm 0.3 \times 10^{-3}$
7.5×10^{-2}	$6.7 \pm 0.1 \times 10^{-2}$	$6.5 \pm 0.1 \times 10^{-2}$	$6.2 \pm 0.1 \times 10^{-2}$	$6.7 \pm 0.3 \times 10^{-3}$
1×10^{-1}	$8.0 \pm 0.1 \times 10^{-2}$	$7.5 \pm 0.1 \times 10^{-2}$	$1.2 \pm 0.3 \times 10^{-1}$	$2.8 \pm 0.1 \times 10^{-2}$

Table 4: Validation and simulation error of our reduction method on the bike-sharing model case study

simulated. However, the remaining population variables and transitions in the PCTMC are either removed or approximately simulated.

As modellers we know that a model is an abstraction of the system in the real world. Thus it inevitably contains some deviation from the real system due to details that are omitted in the abstraction process. Consequently, except for the case of particular safety critical systems, it is generally acceptable to allow some minor noise to be introduced into a model during construction. Taking this perspective a little further, we can consider the transitions and population variables that we removed from the simulation as noise factors which have negligible impact on the evolution of populations of interest.

We have demonstrated the power of our method by applying it to the stochastic simulation of two PCTMC models in the disease spread and public transportation area. The result shows that our method can achieve significant acceleration of stochastic simulation but still retain high statistical accuracy on the dynamics of the target populations.

Acknowledgement

This work was supported by the EU project QUANTICOL, 600708.

References

[1] L. J. Allen, An introduction to stochastic processes with applications to biology, CRC Press, 2010.

[2] M. Spencer, E. Susko, Continuous-time Markov models for species interactions, Ecology 86 (12) (2005) 3272–3278. doi:10.1890/05-0029.

[3] H. Andersson, T. Britton, Stochastic Epidemic Models and Their Statistical Analysis, Springer-Verlag, 2000.

[4] D. F. Anderson, T. G. Kurtz, Continuous time Markov chain models for chemical reaction networks, in: Design and analysis of biomolecular circuits, Springer, 2011, pp. 3–42. doi:10.1007/978-1-4419-6766-4_1.

[5] C. Fricker, N. Gast, Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity, EURO Journal on Transportation and Logistics (2014) 1–31doi:10.1007/s13676-014-0053-5.

[6] M. C. Guenther, J. T. Bradley, Journey data based arrival forecasting for bicycle hire schemes, in: Analytical and Stochastic Modeling Techniques and Applications, Vol. 7984 of LNCS, Springer, 2013, pp. 214–231. doi:10.1007/978-3-642-39408-9_16.

[7] C. Feng, J. Hillston, D. Reijsbergen, Moment-based probabilistic prediction of bike availability for bike-sharing systems, in: International Conference on Quantitative Evaluation of Systems, Springer, 2016, pp. 139–155. doi:10.1007/978-3-319-43425-4_9.

[8] T. G. Kurtz, Approximation of population processes, SIAM, 1981.

[9] L. Bortolussi, J. Hillston, D. Latella, M. Massink, Continuous approximation of collective system behaviour: A tutorial, Performance Evaluation 70 (5) (2013) 317–349. doi:10.1016/j.peva.2013.01.001.

[10] M. C. Guenther, A. Stefanek, J. T. Bradley, Moment closures for performance models with highly non-linear rates, in: European Workshop on Performance Engineering, Springer, 2012, pp. 32–47. doi:10.1007/978-3-642-36781-6_3.

[11] J. Hasenauer, V. Wolf, A. Kazeroonian, F. Theis, Method of conditional moments (MCM) for the chemical master equation, Journal of Mathematical Biology 69 (3) (2014) 687–735. doi:10.1007/s00285-013-0711-5.

[12] D. T. Gillespie, Exact stochastic simulation of coupled chemical reactions, The Journal of Physical Chemistry 81 (25) (1977) 2340–2361. doi:10.1021/j100540a008.

[13] D. T. Gillespie, Stochastic simulation of chemical kinetics, Annu. Rev. Phys. Chem. 58 (2007) 35–55.

[14] Y. Cao, H. Li, L. Petzold, Efficient formulation of the stochastic simulation algorithm for chemically reacting systems, The Journal of Chemical Physics 121 (9) (2004) 4059–4067. doi:10.1063/1.1778376.

[15] M. A. Gibson, J. Bruck, Efficient exact stochastic simulation of chemical systems with many species and many channels, The Journal of Physical Chemistry A 104 (9) (2000) 1876–1889. doi:10.1021/jp993732q.

[16] D. T. Gillespie, Approximate accelerated stochastic simulation of chemically reacting systems, The Journal of Chemical Physics 115 (4) (2001) 1716–1733. doi:10.1063/1.1378322.

[17] Y. Cao, D. T. Gillespie, L. R. Petzold, Efficient step size selection for the tau-leaping simulation method, The Journal of Chemical Physics 124 (4) (2006) 044109. doi:10.1063/1.2159468.

[18] C. V. Rao, A. P. Arkin, Stochastic chemical kinetics and the quasi-steady-state assumption: application to the Gillespie algorithm, The Journal of Chemical Physics 118 (11) (2003) 4999–5010. doi:10.1063/1.1545446.

[19] E. A. Mastny, E. L. Haseltine, J. B. Rawlings, Two classes of quasi-steady-state model reductions for stochastic kinetics, The Journal of Chemical Physics 127 (9) (2007) 094106. doi:10.1063/1.2764480.

[20] Y. Pu, L. T. Watson, Y. Cao, Stiffness detection and reduction in discrete stochastic simulation of biochemical systems, The Journal of Chemical Physics 134 (5) (2011) 054105. doi:10.1063/1.3548838.

[21] Y. Cao, D. T. Gillespie, L. R. Petzold, The slow-scale stochastic simulation algorithm, The Journal of Chemical Physics 122 (1) (2005) 014116. doi:10.1063/1.1824902.

[22] L. Bortolussi, R. Paškauskas, Mean-field approximation and quasi-equilibrium reduction of Markov population models, in: Quantitative Evaluation of Systems, Vol. 8657 of LNCS, Springer, 2014, pp. 106–121. doi:10.1007/978-3-319-10696-0_9.

[23] C. Feng, J. Hillston, Speed-up of stochastic simulation of pctmc models by statistical model reduction, in: European Workshop on Performance Engineering, Springer, 2015, pp. 291–305. doi:10.1007/978-3-319-23267-6_19.

[24] S. Engblom, Computing the moments of high dimensional solutions of the master equation, Applied Mathematics and Computation 180 (2) (2006) 498–515. doi:10.1016/j.amc.2005.12.032.

[25] L. Isserlis, On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables, Biometrika (1918) 134–139doi:10.2307/2331932.

[26] A. Ale, P. Kirk, M. P. Stumpf, A general moment expansion method for stochastic kinetic models, The Journal of Chemical Physics 138 (17) (2013) 174101. doi:10.1063/1.4802475.

[27] M. J. Keeling, Multiplicative moments and measures of persistence in ecology, Journal of Theoretical Biology 205 (2) (2000) 269–281. doi:10.1006/jtbi.2000.2066.

[28] A. Singh, J. P. Hespanha, Lognormal moment closures for biochemical reactions, in: 45th IEEE Conference on Decision and Control, IEEE, 2006, pp. 2063–2068. doi:10.1109/CDC.2006.376994.

[29] I. Krishnarajah, A. Cook, G. Marion, G. Gibson, Novel moment closure approximations in stochastic epidemics, Bulletin of Mathematical Biology 67 (4) (2005) 855–873. doi:10.1016/j.bulm.2004.11.002.

[30] I. Nåsell, An extension of the moment closure method, Theoretical Population Biology 64 (2) (2003) 233–239. doi:10.1016/S0040-5809(03)00074-1.

[31] A. Slepoy, A. P. Thompson, S. J. Plimpton, A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks, The

Journal of Chemical Physics 128 (20) (2008) 205101. doi:10.1063/1.2919546.

[32] S. Wu, J. Fu, H. Li, L. Petzold, Automatic identification of model reductions for discrete stochastic simulation, The Journal of Chemical Physics 137 (3) (2012) 034106. doi:10.1063/1.4733563.

[33] L. Bortolussi, D. Milios, G. Sanguinetti, Efficient stochastic simulation of systems with multiple time scales via statistical abstraction, in: Computational Methods in Systems Biology, Vol. 9308 of LNCS, Springer, 2015, pp. 40–51. doi:10.1007/978-3-319-23401-4_5.

[34] T. Lu, C. K. Law, A directed relation graph method for mechanism reduction, Proceedings of the Combustion Institute 30 (1) (2005) 1333–1341. doi:10.1016/j.proci.2004.08.145.

[35] P. Pepiot-Desjardins, H. Pitsch, An efficient error-propagation-based reduction method for large chemical kinetic mechanisms, Combustion and Flame 154 (1) (2008) 67–81. doi:10.1016/j.combustflame.2007.10.020.

[36] K. E. Niemeyer, C.-J. Sung, M. P. Raju, Skeletal mechanism generation for surrogate fuels using directed relation graph with error propagation and sensitivity analysis, Combustion and Flame 157 (9) (2010) 1760–1770. doi:10.1016/j.combustflame.2009.12.022.

[37] C. Feng, J. Hillston, V. Galpin, Automatic moment-closure approximation of spatially distributed collective adaptive systems, ACM Transactions on Modeling and Computer Simulation (TOMACS) 26 (4) (2016) 26. doi:10.1145/2883608.

[38] E. W. Dijkstra, A note on two problems in connexion with graphs, Numerische mathematik 1 (1) (1959) 269–271.

[39] T. H. Cormen, Introduction to algorithms, MIT press, 2009.

[40] M. Tschaikowski, M. Tribastone, Approximate reduction of heterogenous nonlinear models with differential hulls, IEEE Transactions on Automatic Control 61 (4) (2016) 1099–1104. doi:10.1109/TAC.2015.2457172.

[41] R. K. Watson, On an epidemic in a stratified population, Journal of Applied Probability 9 (03) (1972) 659–666. doi:10.2307/3212334.

[42] A. Bhattachayya, On a measure of divergence between two statistical population defined by their population distributions, Bulletin Calcutta Mathematical Society 35 (99-109) (1943) 28.